\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW6\_1

Author: F74042086郭泰佑 <st9540808@yhaoo.com.tw>

Class: 乙班

Description:

How do you finish this homework?

linux沒有conio.h標頭檔,所以getche()是存網路上抄下來的

using integer pointer.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

//學號：F74042086

//姓名：郭泰佑

//編譯方式：gcc -o hw6\_1 hw6\_1.c -lm

//執行方式：./hw6\_1

//程式功能：convert binary to float or double and vice versa

//更新日期：2015.12.15

#include <stdio.h>

#include <termios.h>

#include <unistd.h>

#include <math.h>

char getche()//copy from stackoverflow

{

char buf=0;

struct termios old={0};

fflush(stdout);

if(tcgetattr(0, &old)<0)

perror("tcsetattr()");

old.c\_lflag&=~ICANON;

old.c\_lflag&=~ECHO;

old.c\_cc[VMIN]=1;

old.c\_cc[VTIME]=0;

if(tcsetattr(0, TCSANOW, &old)<0)

perror("tcsetattr ICANON");

if(read(0,&buf,1)<0)

perror("read()");

old.c\_lflag|=ICANON;

old.c\_lflag|=ECHO;

if(tcsetattr(0, TCSADRAIN, &old)<0)

perror ("tcsetattr ~ICANON");

printf("%c",buf);

return buf;

}

int main()

{

//float

float \*floatVar, input;

printf("Input float number : ");

scanf("%f",&input);

floatVar = &input;

int \*fl = (int\*)floatVar;

int binary[32], i;

//印出float的binary representation

for (i = 0; i < 32; i++) binary[i] = ((1 << i) & \*fl) != 0 ? 1 : 0;

for (i = 31; i >= 0; i--) printf("%d",binary[i]);

printf("\n");

printf("Input binary number to convert to float :\n");

for (i = 0; i < 32; i++)//每按一個鍵就存入binary陣列中 不用按enter 打錯無法使用backspace

binary[i] = (int)getche() - 48;

printf("\n");

struct sem{

unsigned int mantissa : 23;

unsigned int exp : 8;

unsigned int sign : 1;

}fl\_1, \*ptr;

fl\_1.sign = 0;fl\_1.exp = 0; fl\_1.mantissa = 0;

for (i = 0; i < 32; i++)//把相對應的部份存入bit field

{

if( i == 0 ) fl\_1.sign = ( binary[0] != 0 ? 1 : 0 ) ;

if(( i >= 1 && i <= 8) && binary[i] == 1 ) fl\_1.exp += (int)pow(2,8-i);

if(( i >= 9 && i <= 31) && binary[i] == 1 ) fl\_1.mantissa += (int)pow(2,31-i);

}

ptr = &fl\_1;floatVar = (float \*)ptr;//強制轉型

printf("%e\n\n",\*floatVar);//印出float

//double

double \*doubleVar, input\_dou;

printf("Input double number : ");

scanf("%lf",&input\_dou);

doubleVar = &input\_dou;

long long \*dou = (long long\*)doubleVar;

int binary\_dou[64];

for (i = 0; i < 64; i++) binary\_dou[i] = (((long long)1 << i) & \*dou) != 0 ? 1 : 0;

for (i = 63; i >= 0; i--) printf("%d",binary\_dou[i]);

printf("\n");

printf("Input binary number to convert to double :\n");

for (i = 0; i < 64; i++) binary\_dou[i] = (int)getche() - 48;

printf("\n");

struct sem\_dou{

unsigned long long mantissa : 52;

unsigned int exp : 11;

unsigned int sign : 1;

}dou\_1, \*ptr\_dou;

dou\_1.sign = 0;dou\_1.exp = 0; dou\_1.mantissa = 0;

for (i = 0; i < 64; i++)//把相對應的部份存入bit field

{

if( i == 0 ) dou\_1.sign = ( binary\_dou[0] != 0 ? 1 : 0 ) ;

if(( i >= 1 && i <= 11) && binary\_dou[i] == 1 ) dou\_1.exp += (int)pow(2,11-i);

if(( i >= 12 && i <= 63) && binary\_dou[i] == 1 ) dou\_1.mantissa += (long long)pow(2,63-i);

}

ptr\_dou = &dou\_1;doubleVar = (double \*)ptr\_dou;

printf("%e\n",\*doubleVar);

return 0;

}

Compilation:

gcc -o hw6\_1 hw6\_1.c -lm

Execution:

./hw6\_1

Output:

F74042086@c-2015-1:~/hw6> ./hw6\_1

Input float number : -2.5

11000000001000000000000000000000

Input binary number to convert to float :

11000000001000000000000000000000

-2.500000e+00

Input double number : -3.5

1100000000001100000000000000000000000000000000000000000000000000

Input binary number to convert to double :

1100000000001100000000000000000000000000000000000000000000000000

-3.500000e+00

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW6\_2

Author: F74042086郭泰佑 <st9540808@yhaoo.com.tw>

Class: 乙班

Description:

using union.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

//學號：F74042086

//姓名：郭泰佑

//編譯方式：gcc -o hw6\_2 hw6\_2.c -lm

//執行方式：./hw6\_2

//程式功能：convert binary to float or double and vice versa

//更新日期：2015.12.15

#include <stdio.h>

#include <termios.h>

#include <unistd.h>

#include <math.h>

char getche()//copy from stackoverflow

{

char buf=0;

struct termios old={0};

fflush(stdout);

if(tcgetattr(0, &old)<0)

perror("tcsetattr()");

old.c\_lflag&=~ICANON;

old.c\_lflag&=~ECHO;

old.c\_cc[VMIN]=1;

old.c\_cc[VTIME]=0;

if(tcsetattr(0, TCSANOW, &old)<0)

perror("tcsetattr ICANON");

if(read(0,&buf,1)<0)

perror("read()");

old.c\_lflag|=ICANON;

old.c\_lflag|=ECHO;

if(tcsetattr(0, TCSADRAIN, &old)<0)

perror ("tcsetattr ~ICANON");

printf("%c",buf);

return buf;

}

//宣告union包含 float&double 的 bit field

union sem{

struct sem\_float{

unsigned int mantissa : 23;

unsigned int exp : 8;

unsigned int sign : 1;

}fl\_part;

struct sem\_double{

unsigned long long mantissa : 52;

unsigned int exp : 11;

unsigned int sign : 1;

}dou\_part;

}binarycode, \*ptr;

int main()

{

//float

float \*floatVar, input;

printf("Input float number : ");

scanf("%f",&input);

floatVar = &input;

int \*fl = (int\*)floatVar;

int binary[32], i;

//印出float的binary representation

for (i = 0; i < 32; i++) binary[i] = ((1 << i) & \*fl) != 0 ? 1 : 0;

for (i = 31; i >= 0; i--) printf("%d",binary[i]);

printf("\n");

printf("Input binary number to convert to float :\n");

for (i = 0; i < 32; i++)//每按一個鍵就存入binary陣列中 不用按enter 打錯無法使用backspace

binary[i] = (int)getche() - 48;

printf("\n");

binarycode.fl\_part.sign = 0;

binarycode.fl\_part.exp = 0;

binarycode.fl\_part.mantissa = 0;

for (i = 0; i < 32; i++)//把相對應的部份存入bit field

{

if( i == 0 ) binarycode.fl\_part.sign = ( binary[0] != 0 ? 1 : 0 ) ;

if(( i >= 1 && i <= 8) && binary[i] == 1 )

binarycode.fl\_part.exp += (int)pow(2,8-i);

if(( i >= 9 && i <= 31) && binary[i] == 1 )

binarycode.fl\_part.mantissa += (int)pow(2,31-i);

}

ptr = &binarycode;floatVar = (float \*)ptr;//強制轉型

printf("%e\n\n",\*floatVar);//印出float

//double

double \*doubleVar, input\_dou;

printf("Input double number : ");

scanf("%lf",&input\_dou);

doubleVar = &input\_dou;

long long \*dou = (long long\*)doubleVar;

int binary\_dou[64];

for (i = 0; i < 64; i++) binary\_dou[i] = (((long long)1 << i) & \*dou) != 0 ? 1 : 0;

for (i = 63; i >= 0; i--) printf("%d",binary\_dou[i]);

printf("\n");

printf("Input binary number to convert to double :\n");

for (i = 0; i < 64; i++) binary\_dou[i] = (int)getche() - 48;

printf("\n");

binarycode.dou\_part.sign = 0;

binarycode.dou\_part.exp = 0;

binarycode.dou\_part.mantissa = 0;

for (i = 0; i < 64; i++)//把相對應的部份存入bit field

{

if( i == 0 ) binarycode.dou\_part.sign = ( binary\_dou[0] != 0 ? 1 : 0 ) ;

if(( i >= 1 && i <= 11) && binary\_dou[i] == 1 )

binarycode.dou\_part.exp += (int)pow(2,11-i);

if(( i >= 12 && i <= 63) && binary\_dou[i] == 1 )

binarycode.dou\_part.mantissa += (long long)pow(2,63-i);

}

ptr = &binarycode;doubleVar = (double \*)ptr;

printf("%e\n",\*doubleVar);

return 0;

}

Compilation:

gcc -o hw6\_2 hw6\_2.c -lm

Execution:

./hw6\_2

Output:

F74042086@c-2015-1:~/hw6> ./hw6\_2

Input float number : -23.3

11000001101110100110011001100110

Input binary number to convert to float :

11000001101110100110011001100110

-2.330000e+01

Input double number : -33.5

1100000001000000110000000000000000000000000000000000000000000000

Input binary number to convert to double :

1100000001000000110000000000000000000000000000000000000000000000

-3.350000e+01

2-1

yes, by putting 00000000100000000000000000000000

into the “binary to float ” section. We get 0.000000000000000000000000000000000000011754943508222875079687365372222456778186655567720875215087517062784172594547271728515625000000000000000000000000

with 150 precision.

Also putting the value into code like this :

float \*floatVar, input;

input = 1.175494350822287507968736537222245677818665556772087521508751706278417259454727172851560500000000000000000000000000000000e-38f;

floatVar = &input;

int \*fl = (int\*)floatVar;

int binary[32], i;

for (i = 0; i < 32; i++) binary[i] = ((1 << i) & \*fl) != 0 ? 1 : 0;

for (i = 31; i >= 0; i--) printf("%d",binary[i]);

we get 00000000100000000000000000000000 in the output.

Thus, we can regard the value the smallest positive float number.

2-2

bit pattern of 0.0 is

00000000000000000000000000000000

2-3

output 結果為 f1 = f2

因為精度的關係 ,小數點越後面的數字越不重要,所以電腦才會認為兩個值相等